



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/640,620	08/12/2003	Arra E. Avakian	10017134-1	1119

22879 7590 02/26/2007
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER

VU, TUAN A

ART UNIT PAPER NUMBER

2193

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/26/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No. 10/640,620	Applicant(s) AVAKIAN ET AL.	
	Examiner Tuan A. Vu	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 28 November 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>11/28/06</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/28/06.

As indicated in Applicant's response, claims 1, 15, 20 have been amended. Claims 1-20 are pending in the office action.

Claim Objections

2. Claim 20 is objected to because of the following informalities: the reciting of 'instructions operable by a computer *cause* the computer to perform ...' does not make it clear whether the instructions when executed cause the computer to perform, or the computer medium causes the computer to perform. The claim lacks some linking conjunction to set forth the context as to which entity is causing the computer to perform; e.g. '...instructions operable by a computer which when executed cause the computer to perform ...'

Appropriate correction is required.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 20 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a "useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

As per claim 20, this claim recites computer medium storing instructions operable by a computer, which when executed cause the computer to perform a method for instrumenting a runtime chain of transactions and for generating correlators that identify top level transaction and parent corresponding to a transaction. As recited, there is execution by computer instructions for a purpose of instrumenting and creating correlators. That is, the instructions as recited amount to mere execution setting in which functionality with an intended use is to be expected, and absent any actual step taken to carry out the intended use otherwise referred to as 'for instrumenting' and 'for generating correlators', it is unclear that such functionality can be realized in terms of data transformation, or to materialize the use of the method being performed by the computer (*instrumenting, generating correlators to identify*) in terms of a result set forth by the above Practical Application Test. Lacking a reasonable teaching about step actions taken to actually yield some real-world application results being concrete and useful, the method thus recited amounts to claiming an computer-based outset for performing a method with some mere intended use. The claim amounts to establishing of a preamble without providing a body of a main claim.

In short, intended functionality claimed without reasonable teaching of steps taken to actualize (via some action being performed or data transformation) such functionality in specific concrete actions in order to yield a Practical Application result, the claim fails to provide 'useful, concrete and tangible result' and is rejected for leading to non-statutory subject matter.

Claim Rejections - 35 USC § 102

Art Unit: 2193

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claim 20 is rejected under 35 U.S.C. 102(e) as being anticipated by Aman et al.,

USPubN: 2004/0220947 (hereinafter Aman).

As per claim 20, Aman discloses a computer readable medium for storing instructions operable by a computer, which when executed cause the computer to perform a method for instrumenting at run-time (e.g. *dynamically, ARM, API, instrumentation interfaces* - para 0126-0128, pg. 7-8; Fig. 4, Fig. 12) a hierarchical chain of parent-child transactions including a top level transition and at least one child transaction thereof (e.g. Fig. 26; para 0021-0024, pg. 2; *end-to-end, parent transaction, nested transactions* - para 0191-0195, pg 11) without modifying source codes (e.g. Fig. 4 – Note: dynamic insertion of API call via ARM services reads on without modifying source code of transactions – see Fig. 12) associated with these transactions and for generating correlators for each of said transactions (e.g. correlator -Fig. 26; para 0135-0136, pg. 8), wherein each correlator identifies said top level transaction and a parent transaction, if any, corresponding to its associated transaction (e.g. Fig. 26; para 0021-0024, pg. 2).

7. Claims 1-12 are rejected under 35 U.S.C. 102(e) as being anticipated by Labadie et al, USPubN: 2003/0195959 (hereinafter Labadie)..

As per claim 1, Labadie discloses a server system method for monitoring performance of a plurality of transactions including a top level transaction and plurality of transactions relating

Art Unit: 2193

to said top level transaction in a child parent hierarchy (e.g. Tivoli ARM ... International Business Machines - para 0005-0014, pg. 1-2; para 0036, pg. 4; *event originated; event that triggered the particular event* - para 0045-0052, pg. 4-5 – Note: ARM and Tivoli correlators reads on parent/child transaction correlators with associated measurements via API, correlation that identifying originating or triggering events/host name), comprising

for each of selected ones of said transactions, instrumenting said transaction at run-time without modifying its source code (e.g. Fig. 4A-C- Note: Middleware instrumenting of live events and transaction threads reads on without modifying corresponding web code) to obtain a performance metric corresponding thereto (e.g. para 0061, pg 5; Fig. 5A, 5B),

for each of said instrumented transactions, generating a correlator for identifying said top level transaction and a parent transaction (Note: ARM correlator reads on child/parent relationship), if any, of said transaction, and

utilizing said correlators to cross-correlate a performance metric corresponding to a parent transaction with one or more performance metrics corresponding to one or more child transactions of said parent transaction (e.g. Fig. 5B; *SOAP parameters, timestamp* – para 0073, pg. 7; Fig. 6A-C).

Labadie specifies that the server system is a EJB server with applications (para 0026, pg. 3) involving Sun Microsystems Enterprise beans; hence has disclosed that this server system is a J2EE because of the transaction-related ARM services and instrumentation on EJB Java objects (see Fig. 6A-C).

As per claim 2, Labadie discloses the step of instrumenting said transaction comprises inserting instrumentation code in a bytecode representation of said transaction (byte stream – para 0072, pg. 6).

As per claims 3-6, Labadie discloses wherein said performance metric corresponds to a response time of said transaction (*response time* - para 0005-0014, pg. 1-2); wherein said instrumentation code effects generation of a start time marker upon start of execution of said transaction and generation of a stop time marker upon completion of execution of said transaction (para 0068, pg. 6); wherein said instrumentation code generates calls to an Application Response Measurement (ARM) agent to cause generation of said stop and start time markers (service 350 – Fig. 5B; para 0005-0014, pg. 1-2) utilizing said start and stop time markers to measure a response time of said transaction (Fig. 5A, 6A).

As per claim 7, Labadie discloses generating a record for each instrumented transaction upon completion of said transaction, said record indicating said performance metric associated with said transaction (Fig. 5A-B), a parent of said transaction, and said top level transaction (Note: for each byte stream being instrumented for a ARM code instrumentation as disclosed, the top level or correlated event being monitored reads on parent or top level transaction – see para 0045-0052, pg. 4-5).

As per claim 8, Labadie discloses transmitting said transaction record to an analysis and presentation module (e.g. *PushCorrelator*, *GetAllCorrelator* - Fig. 6B; *Set_Context_Data*, *Set_Context_Info* - Fig. 6A, B; *CorrelatorTableEntry* 390 – Fig. 5B).

As per claims 9-10, Labadie discloses storing of said correlators in a thread local storage stack (e.g. Fig. 4A-C - Java Virtual Machine runtime thread with Thread counter reads on thread

Art Unit: 2193

stack in JVM, stack being inherent to a JVM runtime as evidenced by *PushCorrelator*, *PullCorrelator* – Fig. 5B) in case of execution of said hierarchical transactions in a single thread (para 0037-0045, pg. 4); and storing said correlators in the stack based on a LIFO protocol (see Fig. 4A-C - Note: Java Virtual Machine runtime stack for threads recording with inclusion of associated correlators therein at runtime, reads on LIFO protocol of a given stack).

As per claim 11, Labadie discloses removing a correlator from said stack upon completion of a transaction associated with said correlator (*PullCorrelator* – Fig. 6B).

As per claim 12, Labadie discloses wherein said top-level transaction is initiated in response to a request received from a web server (e.g. *Init_Correlator* – Fig. 6A- Note: any partner event in the flow of threads – see Fig. 4A-C- reads on a top thread leading to other thread downstream based on the first request and *Init* – see *InitClientMWare* –Fig. 6A, in light para 0034, pg. 4).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claim 13, 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Labadie et al, USPubN: 2003/0195959 (hereinafter Labadie), and further in view of Hind et al, USPN: 7,003,565 (hereinafter Hind).

As per claims 13-14, Labadie does not explicitly disclose wherein said web server transmits a cookie to said application server together with said request; and further utilizing said cookie to generate said top level correlator. But the client state being collected and passed (see Fig. 5A-B) over to different servers (plug-in middleware, DCS correlator service) using the instrumentation service (ARM) to record correlator as shown by Labadie (see para 0028-0032, 0034-0036) entail client runtime data/events to be passed from services to services to enable correlation of thread or partners being enumerated for a process request or analysis thereof(see Fig. 4A-C). The use of cookie at a given machine to store client data for repeated usage – so to obviate creation of additional discovery resources -- was concept used in the data collecting paradigm by Hind so that by using these record or cookie under the provision of message as to communicate with servers (see Fig. 3A; correlator – col. 8, lines 20-67) Hind's collection of correlator-type of data can support service as to improve QoS delivery or administrative policy enforcing. Hence, in the same endeavor as analyzing performance of transaction as Labadie, Hind provides in-bound and out-bound message with cookie data so to provide very specific client data for server to enforce quality control transaction using correlation information therein (see Fig. 6) thus to alleviate dependency of information interchanges from many sources of data providers (or linked servers) as cookie messaging can yield latest state of client information. Hence, in view to the multiple agent messaging as required in Labadie's correlator record passing, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide cookie messaging by Hind, i.e. using said cookie data to implement or to support correlator data collection as purported by Labadie. One skill in the art would be motivated to do so because cookie data from one sending edge server to the next would alleviate

Art Unit: 2193

extraneous discovery resources for these data reflect the most accurate and dynamic state of the client information being passed (see Hind, col. 16, bottom to col 17 line 34) such that by utilizing this cookie approach, the servers can make use of the most up-to-date state of a client/requesting source data to fulfill the quality of transaction as approached by Labadie's instrumentation service, or to facilitate the enforcement of transaction security as endeavored by Hind's Qos paradigm.

10. Claim 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Labadie et al, USPubN: 2003/0195959 (hereinafter Labadie), and further in view of Bansal et al., USPubN: 2003/0120593 (hereinafter Bansal)

As per claim 15, Labadie discloses a method for monitoring performance of at least two Java transactions executing in two separate processes and being related to one another as parent-child transactions (re claim 1), comprising instrumenting each of said transactions at run-time by modifying its respective bytecode representation (e.g. *SOAP*, *jvmID* – para 0036-0044, pg. 4; *JMS*, para 0049, pg. 4; *toString* – para 0053, pg. 5; Fig. 3, 5, 6 – Note: threads in a java-based event capturing *SOAP*, using instances of correlator's methods – e.g. *toString* -- and JVM runtime entails execution of code by modifying a runtime JVM bytecodes) to obtain a selected performance metric corresponding thereto, generating a correlator corresponding to said top-level transaction.

Labadie discloses utilizing RMI (see ORB, para 0028, pg. 3) to send said top-level correlator incorporated in a header of an IIOP message to said child transaction, and generator another correlator corresponding to said child transaction (Fig. 5A-B; *header* - para 0064-0066, pg. 6); but does not explicitly teach RMI over IIOP. The use of message over IIOP in a J2EE

Art Unit: 2193

based network is taught by Bansal (Fig. 23) who also teaches using of ARM to instrument and measure application data for performance reporting (see para 0922-0969, pg. 38-40). Since Labadie is also suggesting performance analysis in a similar context where message containing correlator data are passed in a interoperability Enterprise Java network (para 0026, pg. 3), it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a layer of IIOP as in Bansal's message passing above among the ORB layer pertinent to this J2EE paradigm in order for Labadie's RMI invocation (over ORB) or correlator record passing to benefit of the core service of the ORB based on IIOP as heterogeneous format data can be reconverted from one format into another to fulfill the path of the data being transferred in this enterprise communications means.

As per claims 16-19, these claims include the subject matter of claims 3-5 or 6; hence will incorporate the corresponding rejection as set forth therein, respectively.

Response to Arguments

11. Applicant's arguments filed 11/28/06 have been fully considered but they are not persuasive. Following are the Examiner's observations in regard thereto.

Rejection under 35 USC §101:

(A) The changes made to claim 20 fall under a deficiency that is now explained as merely setting a method execution with **two intended uses** without specific method steps taken in order to convey that **data is being transformed** in order to fulfill the Practical Test Application requirements in regard to generating a concrete, useful and tangible result.

Rejection under 35 USC §102 and §103:

Art Unit: 2193

(B) Applicants have submitted (for claim 1) that Labadie's threads are already instrumented prior to the events illustrated by Fig. 4A; and that Labadie's *logging provider* teaches invocations by the provider to selectively capture events of interest (Appl. Rmrks, pg. 9, middle); and those do not disclose 'instrumenting a transaction at run-time without modifying its source code' as claimed. The claimed limitation at stakes here relies on 2 main concepts: (i) instrumenting without modifying a transaction's source code; and (ii) doing so at runtime of the transaction. The claim however does not provide sufficient details as to what *instrumenting* amounts to in terms of steps taken to implement what is claimed as a metric obtaining function. Not modifying a source code is but one aspect of instrumenting a program, but is not the only way to construe instrumenting. Instrumentation is rather viewed as a method of implementing (via addition/inserting of hardware or software-based tools or functions) instructions invocations or active modules within the execution path of a program or a running entity (which can include more sub-entities) in order to collect performance or events-related data or metrics for some subsequent analysis. The adding or inserting of means to collect the metrics can be done before, just before, or during runtime of the program; and in most of runtime type of instrumentation, source code modification/insertion would not be reasonably applicable. Labadie teaches instrumenting without modifying a source code; thus teaches doing this during runtime. That is, Labadie invokes call via API of a ARM service, and this type fits to the paradigm (i), in light of known and accepted meaning of a program instrumentation as mentioned above; and doing it, Labadie has been able to provide the *correlators* function to assist the analysis of transaction between the computers (see para 0005, pg. 1, R column), e.g. measurement during such transaction, thus runtime as in (ii). The argument about threads including pre-instrumented code

Art Unit: 2193

does not correlate with the actual state of the claimed language; and amounts to allegation without pointing out how the language as recited distinguish over any of the cited portions. It is noted that Labadie's use of plug-in, DCS servers in a correlator-based dynamic service is founded upon invoking a runtime plug-in using middleware to obtain instances of correlators (i.e. instance dynamically created, not pre-established inside a transaction source program) and associated partner equivalents (see para 0030-0035). Figure 4s describe points inside of threads that are set up by the middleware or correlator to have metrics captured and this does not entail that there is source code insertion as in a non-runtime source code instrumentation because correlators are in part information structures being passed across requestors. It is actually via the dynamic invocation of plug-in that the DCS server and its middleware agent are be able to provide this correlation service; and by having this dynamic *plug-in* invocation that more methods inside each correlator (see para 0053, pg. 5) can capture threads events as seen in Figures 4. It is also noted that the nature of threads is fundamentally dynamic, and this by itself precludes any remote possibility as to capture source code thereof in order to provide pre-runtime instrumentation code insertion. That is, there is no predesigned insertion of instrumentation source code related to a *transaction* program, rendering the argument about source code pre-insertion not convincing. The claimed *without modifying source code of a transaction*, as interpreted is not different from Labadie's invoking (via a server call) a middleware plug-in to trigger more middleware invocation to obtain a correlator service so that when evoked each correlator can utilize its dynamic (emphasis added) instance to enable more methods (see Fig. 5B) to capture events. Labadie has fulfilled what instrumenting at runtime is: capturing of metrics without modifying a source code; and doing this during runtime of

Art Unit: 2193

transactions between machines. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C) Applicants have submitted that the Examiner appears to rely on RIOP and IOP to provide grounds for rejecting 'instrumenting ... transactions at runtime by modifying ... bytecode ... thereto' (Appl. Rmrks, pg. 10, 2nd para) and this is falling under the same deficiencies about Labadie's teaching as in claim 1. The rejection has set forth how bytecodes at runtime is being modified by way of methods invoked inside each correlator-related calls in light of a JVM environment implicating messages across dynamically executing transactions, wherein correlators being instantiated are having additional methods to invoke the JVM-related execution of threads or to capture threads performance metrics, which is analogous to modifying this JVM *bytecodes* flow; and runtime flow of a JVM necessarily signifies *bytecodes* execution because such transaction code being modified dynamically CANNOT be standard uncompiled source code, according to one skill in the art. Bansal's teaching is to reinforce how the RMI by Labadie can render this IOP limitation obvious. The argument about bytecodes would now considered moot because the *bytecodes* limitation is addressed as set forth in the rejection. The rejection does not cite Bansal separately from the the RMI and JVM context of the correlator-based request in Labadie while the argument appears to focus on just Bansal's IOP or Labadie's bycodes, in separate avenues of rebut. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413,

Art Unit: 2193

208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(D) The arguments concerning claim 20 (Appl. Rmrks, pg. 10, middle, pg. 11, top) raise that Aman's means of DLL to enable workload understanding across multiple platforms only signifies that instrumentation has occurred. This line of argument falls under the ambit of disagreeing with the invocation via dynamic plug-ins by Labadie that has been addressed in claim 1. Instrumenting *without modifying source code* and doing it at runtime have been the core of how claim 1 or claim 15 is all about; and Aman (in the intended use on workload analysis) by invoking dynamic library APIs to collect performance of live and running transactions across machines also entails invoking of calls similar to Labadie, and doing so without modifying any transaction related source program because the mere fact of linking a DLL to an application is evidence that no source program has been inserted of instructions that require recompilation. And the argument using Aman's Fig. 4 for evoking a runtime invocation (ARM by an agent) would also considered non-persuasive in view of the analysis (refer to section B) about how a instrumentation is normally construed as and how the claim language is devoid of specificity as set forth above.

(E) Applicants have submitted that (for claim 20) Aman fails to teach instrumenting a chain of run-time transactions in view of Aman's Figure 12 (Appl. Rmrks, pg. 11, middle) and also does not disclose 'dynamic insertion of API call via ARM services...' as alleged by Examiner. Aman's invoking of an ARM API would be considered invoking to obtain correlator information in the analogous sense that Labadie executes via a DCS invocation of middleware repeated plug-ins to yield information capture results. All of this, in light of the amount of end-to-end and

Art Unit: 2193

nested transactions in the midst of response/request flow (see Aman, para 0192-0195) is deemed sufficient to fulfill the requirements of the claim, because it is by means of invocation so that code executing the performance collecting can be sequentially intervening during the above flow of responses/requests that the ARM service when invoked can effectuate modifying of a multiple machine transaction so to collect the load related information, no source code required to have instructions insertion for recompiling. The argument about Aman is insufficient to overcome the rejection; that is, *hierarchical chain* is disclosed and *instrumenting without modifying* is disclosed.

As a whole, the arguments are non-convincing; and the claims stand rejected as set forth in the Office Action.

Conclusion

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2193

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
February 21, 2007